Procedia Computer Science
Volume 51, 2015, Pages 2804–2808
ICCS 2015 International Conference On Computational Science

A Methodology for Designing Energy-Aware Systems for Computational Science

Pablo C. Cañizares[1], Alberto Núñez[1], Manuel Núñez[1], and Juan J. Pardo[2]

[1] Universidad Complutense de Madrid, Madrid, Spain.
alberto.nunez@pdi.ucm.es, mn@sip.ucm.es
[2] Universidad de Castilla La-Mancha, Albacete, Spain
juanjose.pardo@uclm.es

Abstract

Energy consumption is currently one of the main issues in large distributed systems. More specifically, the efficient management of energy without losing performance has become a hot topic in the field. Thus, the design of systems solving complex problems must take into account energy efficiency. In this paper we present a formal methodology to check the correctness, from an energy-aware point of view, of large systems, such as HPC clusters and cloud environments, dedicated to computational science. Our approach uses a simulation platform, to model and simulate computational science environments, and metamorphic testing, to check the correctness of energy consumption in these systems.

Keywords: Modeling and Simulation of energy efficient systems, Energy consumption, Cloud computing

1 Introduction

During the last years, the most used approach for processing the ever growing amount of data consists in increasing the number of interconnected processors. This solution boosts the overall system performance, by exploiting parallelism of applications, but it demands huge amounts of energy to supply the system. In fact, the speedup provided by these machines comes at an extremely high energy cost.

Several factors motivate the interest in detecting and reducing the main causes of energy consumption, such as carbon footprint reduction [12] and savings in IT electricity bills [11]. *Green computing*, or sustainable computing, has become the focus attention of initiatives such as Green Grid [1], a global consortium dedicated to advancing energy efficiency in data centers and business computing ecosystems. This consortium has even established a metric called *Power Usage Effectiveness* (PUE) [2], which aims to help to measure and compare the energy efficiency of data centers.

The design of systems solving computational science problems is challenging and requires a certain degree of expertise in the field of distributed systems [14, 8]. In addition, saving energy

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015

doi:10.1016/j.procs.2015.05.438

in large-scale systems is a sensitive issue because end-users will not want to suffer a performance loss. Both performance and energy consumption depend on the underlying architecture of the underlying architecture. Due to the high number of devices that have a direct impact on the overall performance and energy consumption (e.g. CPUs, disks, communication networks, etc), building a system that optimizes such features is a very difficult task.

In this paper we propose a methodology for checking the correctness, from an energy-aware point of view, of complex systems, in particular, those focussing on solving computational science problems. Currently, *testing* [13] is the most widely used technique to validate the correctness of systems. Unfortunately, applying conventional testing techniques for checking complex architectures, such as HPC clusters and cloud computing environments, entails two main difficulties. First, the test suites used to validate these systems contain a large number of tests, which requires massive computing time because each test must be executed in the platform under test. Also, this process requires access to the hardware platform where these tests are executed. Second, an oracle that indicates if a given system is correct or not is, in most situations, unavailable or computationally too expensive [20]. In order to alleviate these difficulties we use a novel approach that appropriately combines two orthogonal techniques: simulation [4, 15] and metamorphic testing [21].

The methodology presented in this paper focuses on analyzing whether energy consumption of a specific complex system is efficient. In order to accomplish this process, we provide the basics to define a whole family of MRs exclusively dealing with the energy consumption of these systems. Therefore, we will be able to check whether a certain architecture is efficient with respect to energy consumption and will be able to choose between different alternatives taking into account whether the priority is on performance or on saving energy.

The rest of the paper is structured as follows. Section 2 shows related work. Section 3 presents an overview of MT and the schema used to define MRs oriented to solve complex (computational science) problems. Finally, section 4 shows our conclusions and some directions for future work.

2 Related Work

In this section we briefly review previous work related to this paper. During the last decade, the research community has developed simulation tools to model and analyze energy consumption of distributed systems. Some of these approaches are CloudSim [3], GreenCloud [7] and CloudExp [10].

It is particularly relevant the work using CloudSim [3]. This tool was initially based on a grid simulator [16]. A new layer was implemented on top of GridSim to add the possibility of simulating cloud systems. Since then, CloudSim has been redesigned from scratch, and does not rely on GridSim anymore.

GreenCloud is an extension of the NS2 network simulator [18]. GreenCloud focuses on simulating the communications between processes running in a cloud at packet level. In the same way as NS2, it is written in C++ and OTcl.

CloudExp is a modeling and simulation environment for cloud computing that can be used to evaluate a wide spectrum of cloud components such as processing elements, data centers, storage, networking, Service Level Agreement (SLA) constraints, web-based applications, Service Oriented Architecture (SOA), virtualisation, management and automation, and Business Process Management (BPM).

Although these simulators allow their users to model and simulate energy consumption of a wide variety of systems, the testing process must be manually performed. These tools do not provide mechanisms to (semi-)automatically check the correctness of the energy consumption of the modeled system.

During the last years Metamorphic Testing [6, 21] has been used in very different application domains such as web applications [5] and middleware [19]. This versatility led to us to apply MT to analyze energy consumption in distributed systems, such as HPC clusters and cloud environments. To the best of our knowledge, MT has not been used in the past for this purpose.

There is work aimed to improve energy efficiency by optimizing the application to be executed, instead of properly customizing the underlying architecture of the system [17]. The authors propose a strategy to achieve the optimal energy savings for distributed matrix multiplication via algorithmically trading more computation and communication at a time adaptively with user-specified memory costs for less DVFS switches. This solution saves 7.5% more energy on average than a classic strategy.

3 Energy-aware metamorphic relations

Conventional testing methods require checking whether the output(s) returned by the system under test are the expected ones or not. Schematically, let $\mathcal{S}$ be a system, I be the input domain and S be a test selection strategy. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\} \subseteq I$ be the set of tests generated by using S. When these tests are sequentially applied to the system $\mathcal{S}$ we obtain a sequence of outputs $\mathcal{S}(t_1)$, $\mathcal{S}(t_2)$, ..., $\mathcal{S}(t_n)$. Therefore, if we have an oracle, called f, then we find an error if there exists $t \in \mathcal{T}$ such that $\mathcal{S}(t) \neq f(t)$. But, in general, we will have neither a complete specification nor an oracle and the most we can do is to look for evidence that there has been a failure.

In this work we use MT techniques [6, 21] in order to alleviate the *oracle problem* [20] when testing whether the models of the systems generated by end-users are appropriate with respect to certain criteria. MT uses expected properties of the target functions relating multiple test inputs/observed outputs obtained from the tested system. Since we use simulation to perform the testing process, we use the term *tenant* to refer to a user modeled in the system. A single test case is represented by a set of tenants T executed over a single system model m, denoted by $T(m)$. The idea is to generate *variants* of the original model, compute the application of the set of tenants to these variants, $T(m'_1), T(m'_2), \ldots, T(m'_k)$, and compare the obtained results. Let us remark that our variants are not *mutants* in the sense of *mutation testing* [9]: our goal is not to *kill* the variants but to compare the different obtained results to detect a wrong or suboptimal behavior of the original model. Next, we formally define the pattern of our relations.

Definition 1. Let m be a model that represent the system to be tested and $\mathcal{M}$ be a set of models such that $m \notin \mathcal{M}$. Let T be a set of tenants. A *metamorphic relation* MR for m and T is the set of 5-tuples

$$MR(m,T) = \{(T, m, m', T(m), T(m'))\,|m' \in \mathcal{M} \wedge p_1(m,m') \Rightarrow p_2(m,m',T(m),T(m'))\,\}$$

where p_1 is a relation over models and p_2 is a relation over the models and the execution of tenants on these models. If m and T are fixed then we will write MR instead of $MR(m,T)$. □

HPC clusters and cloud environments contain a vast number of nodes interconnected through a high-speed communication network, where a big number of users concurrently launch applications. Therefore, in order to test the appropriate behavior of these systems we cannot use standard testing techniques. In addition to the oracle problem, that is, we usually do not have an effective method to decide whether the application of a test returned the expected result, we have that standard testing methods do not scale well and we are confronting systems with a huge number of possible configurations. In order to deal with these inherent difficulties, we use MT techniques and a simulation platform to model and simulate the architecture of the system to be tested. Metamorphic testing partially alleviates the oracle problem while simulation allows to compute approximate results, with different configurations and workloads, without building the actual system.

We compare two models by using one set of tenants, denoted by m, m' and T, respectively, where m represents the original model provided by the user, m' represents an automatically generated variant and T represents the workload executed in each model. If a tuple does not belong to an MR then we can say that we found an error. In other words, given a model m that we are validating and a set of tenants T, if we have that for some model m' the tuple $(T, m, m', T(m), T(m'))$ does not belong to MR then the application of these tenants shows that there has been an unexpected behavior. For example, we might expect the performance of m to be always better than the one of m' but, for our chosen set of tenants T, the running of T with m and m' shows otherwise.

In order to define MRs targeted to deal with energy consumption, we propose the following notation: $\Delta(m_{cpu})$ denotes the overall CPU performance of m, $\Delta(m_{IO})$ denotes the overall I/O performance of m, $\Omega(T(m))$ denotes the overall energy consumption required to execute the workload T over m, $\Gamma(T(m))$ denotes the time required to execute the workload T over m, and finally, $|m|$ denotes the number of machines used in m.

4 Conclusions and future work

In this paper we have presented a novel approach that appropriately combines two orthogonal techniques, simulation and metamorphic testing, for checking the correctness of distributed systems with respect to energy consumption. Although this approach seems to be promising, it presents two main limitations. First, the MRs must be provided by the user. Although adding new MRs can be easily performed, this process must be done by hand before executing the testing process. Second, the solution provided by analysing the results is not a global solution. Consequently, the result is heavily dependent on the workload executed in the simulated environment.

As future work we plan to include a new dimension to analyse systems by using MRs: economic cost.

Acknowledgements

Research partially supported by the Spanish MEC project ESTuDIo (TIN2012-36812-C02), the Comunidad de Madrid project SICOMORo-CM (S2013/ICE-3006) and the UCM-Santander program to fund research groups (group 910606).

References

[1] The Green Grid. http://www.thegreengrid.org/ (January 2015), 2015.

[2] C. Belady, A. Rawson, and D. Pfleuger. *Green grid data center power efficiency metrics: PUE and DCiE.* The Green Grid, 2008.

[3] R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In *7th High Performance Computing and Simulation Conference, HPCS'09*, pages 1–11. IEEE Computer Society, 2009.

[4] G. Castañé, A. Núñez, P. Llopis, and J. Carretero. E-mc^2: A formal framework for energy modelling in cloud computing. *Simulation Modelling Practice and Theory*, 39:56–75, 2013.

[5] W. K. Chan, S. C. Cheung, and K. R.P.H. Leung. A Metamorphic Testing Approach for Online Testing of Service-Oriented Software Applications. *International Journal of Web Services Research*, 4(2):61–81, 2007.

[6] T. Y. Chen, S. C. Cheung, and S. M. Yiu. Metamorphic testing: a new approach for generating next test cases. Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, 1998.

[7] Samee Ullah Khan Dzmitry Kliazovich, Pascal Bouvry. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 62(3):1263–1283, 2012.

[8] R. Filgueira, J. Carretero, D. E. Singh, A. Calderón, and A. Núñez. Dynamic-coMPI: Dynamic optimization techniques for MPI parallel applications. *The Journal of Supercomputing*, 59(1):361–391, 2012.

[9] R. M. Hierons, M. G. Merayo, and M. Núñez. Mutation testing. In Phillip A. Laplante, editor, *Encyclopedia of Software Engineering*, pages 594–602. Taylor & Francis, 2010.

[10] Y. Jararweh, M. Jarrah, M. kharbutli, Z. Alshara, M. Noraden Alsaleh, and M. Al-Ayyoub. CloudExp: A comprehensive cloud computing experimental framework. *Simulation Modelling Practice and Theory*, 49:180–192, 2014.

[11] M. Mazzucco, D. Dyachuk, and R. Deters. Maximizing Cloud Provider's Revenues via Energy Aware Allocation Policies. In *IEEE 3rd Int. Conf. on Cloud Computing, CLOUD'10*, pages 131 –138. IEEE Computer Society, 2010.

[12] F. F. Moghaddam, M. Cheriet, and K. K. Nguyen. Low carbon virtual private clouds. In *IEEE 4th Int. Conf. on Cloud Computing, CLOUD'11*, pages 259–266. IEEE Computer Society, 2011.

[13] G.J. Myers. *The Art of Software Testing.* John Wiley and Sons, 2nd edition, 2004.

[14] A. Núñez, J. Fernández, J. D. García, and J. Carretero. Analyzing scalable high-performance I/O architectures. In *3rd Int. Conf. on Parallel and Distributed Processing Techniques and Applications, PDPTA'08*, pages 631–637, 2008.

[15] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente. iCanCloud: A flexible and scalable cloud infrastructure simulator. *Journal of Grid Computing*, 10(1):185–209, 2012.

[16] A. Sulistio, U. Cibej, S. Venugopal, B. Robic, and R. Buyya. A toolkit for modelling and simulating Data Grids: An extension to GridSim. *Concurrency and Computation: Practice and Experience*, 20(13):1591–1609, 2008.

[17] L. Tan, L. Chen, Z. Chen, Z. Zong, R. Ge, and D. Li. HP-DAEMON: High Performance Distributed Adaptive Energy-efficient Matrix-multiplicatiON. In *14th Int. Conference on Computational Science, ICCS'14, Procedia Computer Science 29*, pages 599–613. Elsevier, 2014.

[18] The Network Simulator, NS-2. Web page at `http://www.isi.edu/nsnam/ns/`. Date of last access: 8th August, 2013.

[19] T. H. Tse, S. S. Yau, W. K. Chan, H. Lu, and T. Y. Chen. Testing context-sensitive middleware-based software applications. In *28th Annual Int. Computer Software and Applications Conference, COMPSAC'04*, pages 458–465. IEEE Computer Society, 2004.

[20] E. J. Weyuker. On testing non-testable programs. *The Computer Journal*, 25(4):465–470, 1982.

[21] Z. Q. Zhou, D. H. Huang, T. H. Tse, Z. Yang, H. Huang, and T. Y. Chen. Metamorphic testing and its applications. In *8th Int. Symposium on Future Software Technology, ISFST'04*, pages 1–6. Software Engineers Association, 2004.